

09/27/00

jc885 U.S. PTO

09-29-00

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
UTILITY PATENT APPLICATION TRANSMITTAL  
UNDER 37 CFR 1.53(b)

<b>Address to:</b>  Box Patent Application  Assistant Commissioner for Patents  Washington, DC 20231	<i>Attorney Docket No.</i>	STL9-2000-0080US1
	<i>Inventor(s)</i>	Curt L. Cotner, et al.
	<i>Express Mail Label No.</i>	EL290558674US
	<i>Filing Date</i>	September 27, 2000

jc542 U.S. PTO  
09/27/00

**Title of Application:** *Method of, System for, and Computer Program Product for Providing Automatic Identification of a Computer Program Code Candidate for Web Deployment or a Stored Procedure*

**Transmitted with the patent application are the following:**

- 23 Page(s) Specification, Claims and Abstract
- 6 Page(s) ☒ Formal drawings
- 3 Page(s) Declaration and Power of Attorney (unsigned)
- 1 Page(s) Assignment of the Invention to International Business Machines Corporation
- 2 Page(s) Information Disclosure Statement (IDS/PTO 1449) (copies of citations not included in number of pages)
- X Copies of IDS citations
- X Return Receipt Postcard (MPEP 503).

**Fee Calculation:**

	Claims		Extra	Rate	Fees
<b>Basic Fee</b>					<b>\$690.00</b>
<b>Total Claims</b>	<b>24</b>	<b>-20 =</b>	<b>4</b>	<b>x \$18.00</b>	<b>\$ 72.00</b>
<b>Independent Claims</b>	<b>3</b>	<b>-3 =</b>	<b>0</b>	<b>x \$78.00</b>	
<b>Multiple Dependent Claims</b>				<b>+\$260.00</b>	
					<b>TOTAL</b>
					<b>\$762.00</b>

- X Please charge my Deposit Account No. 09-0460 in the amount of \$ 762.00. **A duplicate copy of this sheet is attached.**
- X The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account 09-0460. **A duplicate copy of this sheet is attached.**
- X Any filing fees under 37 CFR 1.16 for the presentation of extra claims.
- X Any patent application processing fees under 37 CFR 1.17.

**EXPRESS MAIL CERTIFICATE**

I hereby certify that the above paper/fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated below and is addressed to the Assistant Commissioner for Patents, Washington, DC 20231.

Date of Deposit: September 27, 2000

Person Mailing paper/fee: Jeanette Berry Durbin

Signature: Jeanette Berry Durbin

Respectfully submitted,  
Curt Cotner et al.

Prentiss W. Johnson

Prentiss W. Johnson,  
Registration No. #33,123  
Attorney for Applicant(s)  
Telephone (408) 463-5673  
IBM Corporation  
Intellectual Property Law  
555 Bailey Avenue (J46/G467)  
San Jose, CA 95141-1003

Express mail Label #: EL290558674US

## **SPECIFICATION**

IBM Docket No. STL9-2000-0080US1

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN that We, Curt L. Cotner of Gilroy, California and citizen of the United States, Laurence E. England of Morgan Hill, California and citizen of the United States, Howard J. Glaser of San Jose, California and citizen of the United States, and Howard M. Hess of Evanston, Illinois and citizen of the United States have invented new and useful improvements in

### **METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR PROVIDING AUTOMATIC IDENTIFICATION OF A COMPUTER PROGRAM CODE CANDIDATE FOR WEB DEPLOYMENT OR A STORED PROCEDURE**

of which the following is a specification:

1  
2  
3       **METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR**  
4       **PROVIDING AUTOMATIC IDENTIFICATION OF A COMPUTER PROGRAM**  
5       **CODE CANDIDATE FOR WEB DEPLOYMENT OR A STORED PROCEDURE**  
6  
7  
8  
9  
10

11           A portion of the Disclosure of this patent document contains material which is subject  
12 to copyright protection. The copyright owner has no objection to the facsimile reproduction by  
13 anyone of the patent document or the patent disclosure, as it appears in the Patent and  
14 Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

The present invention relates in general to software re-engineering, and more particularly to identifying computer program code which is a candidate for Web enablement or stored procedures.

### 2. Description of the Related Art

A large enterprise may have a significant volume of computer application code, the enterprise's code inventory. Yet the enterprise's programming skill base to maintain, manage, enhance, or re-implement that code inventory may be dwindling. Relative to re-implementation, not all application programs are candidates for conversion to a Web interface or a database stored procedure. Many, if not most, applications do not possess the necessary characteristics for such a conversion. Discovering those existing opportunities to easily access a function from the Web or convert a function to a database stored procedure may be difficult due to the labor intensive nature of the conventional manual techniques. Conventional methods have failed to provide adequate solutions to these problems. Thus, there is a clearly felt need for a method of, system for, article of manufacture for, and computer program product for identifying computer program code which is a candidate for Web enablement or stored procedures.

## SUMMARY OF THE INVENTION

The present invention comprises a method, system, computer program product, and article of manufacture for identifying computer program code which is a candidate for Web enablement or stored procedures. Source code corresponding to computer program code is scanned and parsed to determine static information concerning the computer program code. The static information is stored in a database. Dynamic information concerning the computer program code during an execution of the computer program code is also collected and stored in the database. Responsive to the static information and dynamic information stored in the database, relationships and dependencies are then developed and stored in the database. The database may then be queried to produce a set of potential candidates of computer program code meeting a constraint of the query. If insufficient candidates are returned by the query, then the query constraint may be relaxed, and the query repeated.

A candidate for re-implementation as a database stored procedure call may be identified by a query searching the database for a portion of the computer program code having static information indicating that the portion of the computer program code contains a number, above a specified first threshold, of calls to a database management system, and having dynamic information indicating that the portion of the computer program code is subject to a number of calls, above a specified second threshold, by another portion of computer program code.

A candidate for re-implementation as a Web-enabling interface call may be identified by a query searching the database for a portion of the computer program code having static information indicating that the portion of the computer program contains a transaction and does not contain screen output related program code.

The placement of this information capture, classification, and relationship/dependency discovery into a database, the Application Knowledge Base, that can be queried may provide significant advantages to a user. For an enterprise, this process may be performed across the

1 entire enterprise, encompassing all of the enterprise's inventory of application programs. The  
2 details of the enterprise's application program inventory may be saved and queried in the  
3 application knowledge base. These details may include system and sub-system specific  
4 information. In addition to queries for identification of candidate computer code, the  
5 application knowledge base may simply be queried for application characteristics.  
6

7 Unlike simple conventional string matching or pattern matching approaches, the present  
8 invention discovers relationships and dependencies within the computer program code. For  
9 example, a COBOL dynamic CALL holding the called routine's name in a variable may be  
10 analyzed by data flow analysis to determine the potential values of the target called routine.  
11 Dynamic information obtained during the execution of the COBOL dynamic CALL may  
12 augment the static information obtained through the static analysis and data flow analysis.  
13

14 One aspect of a preferred embodiment of the present invention identifies computer  
15 program code which is a candidate meeting a query constraint.  
16

17 Another aspect of a preferred embodiment of the present invention identifies computer  
18 program code which is a candidate for a stored procedure.  
19

20 Another aspect of a preferred embodiment of the present invention identifies computer  
21 program code which is a candidate for Web enablement.  
22

23 Another aspect of a preferred embodiment of the present invention scans and parses  
24 source code corresponding to computer program code to determine static information  
25 concerning the computer program code, and stores the static information in a database.  
26

27 Another aspect of a preferred embodiment of the present invention collects dynamic  
28 information concerning the computer program code during an execution of the computer  
29 program code, and stores the dynamic information in the database.

1 Another aspect of a preferred embodiment of the present invention discovers or  
2 develops relationships and dependencies responsive to the static information and dynamic  
3 information stored in the database, and stores the relationships and dependencies in the  
4 database.

5  
6 Another aspect of a preferred embodiment of the present invention queries the database  
7 to produce a set of potential candidates of computer program code meeting a constraint of the  
8 query. If insufficient candidates are returned by the query, then the query constraint may be  
9 relaxed, and the query repeated.

10  
11 Another aspect of a preferred embodiment of the present invention identifies a  
12 candidate for re-implementation as a database stored procedure call by a query searching the  
13 database for a portion of the computer program code having static information indicating that  
14 the portion of the computer program code contains a number, above a specified first threshold,  
15 of calls to a database management system, and having dynamic information indicating that the  
16 portion of the computer program code is subject to a number of calls, above a specified second  
17 threshold, by another portion of computer program code.

18  
19 Another aspect of a preferred embodiment of the present invention identifies a  
20 candidate for re-implementation as a Web-enabling interface call by a query searching the  
21 database for a portion of the computer program code having static information indicating that  
22 the portion of the computer program contains a transaction and does not contain screen output  
23 related program code.

1 A preferred embodiment of the present invention has the advantage of promoting reuse  
2 of an existing application program's business logic, resources, or other assets.

3  
4 A preferred embodiment of the present invention has the advantage of promoting  
5 impact analysis and application understanding of an existing application program.

6  
7 A preferred embodiment of the present invention has the further advantage of  
8 promoting reuse of an existing application program's business logic to link the existing  
9 business logic with the Web via a connector to form a new Web application program.

10  
11 A preferred embodiment of the present invention has the further advantage of providing  
12 improved analysis, identification, and isolation of the business logic of an application program.

13  
14 A preferred embodiment of the present invention has the further advantage of reducing  
15 or eliminating labor intensive efforts in the creation of a Web connector through the analysis,  
16 identification, and isolation of the business logic.

17  
18 A preferred embodiment of the present invention has the further advantage of  
19 promoting the use of an enterprise's existing programming skills.

20  
21 A preferred embodiment of the present invention has the further advantage of providing  
22 an application knowledge base, a database containing static information, dynamic information,  
23 and relationships and dependencies of computer program code, which may be queried to  
24 identify candidate computer program code.

25  
26 A preferred embodiment of the present invention has the further advantage of providing  
27 identification of computer program code which is a candidate for stored procedures.

28  
29 A preferred embodiment of the present invention has the further advantage of providing



1 identification of computer program code which is a candidate for Web enablement.

2  
3 A preferred embodiment of the present invention has the further advantage of providing  
4 an improved user interface for identification of computer program code which is a candidate  
5 for Web enablement or stored procedures.

6  
7 A preferred embodiment of the present invention has the further advantage of providing  
8 improved usability in a tool for identification of computer program code which is a candidate  
9 for Web enablement or stored procedures.

10  
11 A preferred embodiment of the present invention has the further advantage of providing  
12 improved functionality in a tool for identification of computer program code which is a  
13 candidate for Web enablement or stored procedures.

## BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the Description of the Preferred Embodiment in conjunction with the attached Drawings, in which:

**Figure 1** is a block diagram of a distributed computer system used in performing the method of the present invention, forming part of the apparatus and computer program product of the present invention, and which may use the article of manufacture and computer program product comprising a computer-readable storage medium having a computer program embodied in said medium which may cause the computer system to practice the present invention;

**Figure 2** is a block diagram of a preferred embodiment of the present invention;

**Figure 3** illustrates a preferred embodiment of the static scanner or parser portion of the preferred embodiment of the present invention;

**Figure 4** illustrates a preferred embodiment of the dynamic scanner or execution monitor parser portion of the preferred embodiment of the present invention;

**Figure 5** illustrates a preferred embodiment of the relationship analyzer portion of the preferred embodiment of the present invention; and

**Figure 6** is a flowchart illustrating the operations preferred in carrying out the preferred embodiment of the present invention.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

An embodiment of the invention is now described with reference to the figures where like reference numbers indicate identical or functionally similar elements. Also in the figures, the left most digit of each reference number corresponds to the figure in which the reference number is first used. While specific configurations and arrangements are discussed, it should be understood that this is done for illustrative purposes only. A person skilled in the relevant art will recognize that other configurations and arrangements can be used without departing from the spirit and scope of the invention. It will be apparent to a person skilled in the relevant art that this invention can also be employed in a variety of other devices and applications.

Referring first to **Figure 1**, there is depicted a graphical representation of a data processing system **8**, which may be utilized to implement the present invention. As may be seen, data processing system **8** may include a plurality of networks, such as Local Area Networks (LAN) **10** and **32**, each of which preferably includes a plurality of individual computers **12** and **30**, respectively. Alternatively, networks **10** and **32** may be intranets or portions of the internet. Of course, those skilled in the art will appreciate that a plurality of Intelligent Work Stations (IWS) coupled to a host processor may be utilized for each such network. Each said network may also consist of a plurality of processors coupled via a communications medium, such as shared memory, shared storage, or an interconnection network. As is common in such data processing systems, each individual computer may be coupled to a storage device **14** and/or a printer/output device **16** and may be provided with a pointing device such as a mouse **17**.

The data processing system **8** may also include multiple mainframe computers, such as mainframe computer **18**, which may be preferably coupled to LAN **10** by means of communications link **22**. The mainframe computer **18** may also be coupled to a storage device **20** which may serve as remote storage for LAN **10**. Similarly, LAN **10** may be coupled via communications link **24** through a sub-system control unit/communications controller **26** and

1 communications link **34** to a gateway server **28**. The gateway server **28** may be an IWS which  
2 serves to link LAN **32** to LAN **10**. Preferably, server **28** is a web application server which  
3 passes transactions from a requester **30** on the internet **32** to the mainframe **18** upon which a  
4 back-end application serving the transaction is executing.

5  
6 With respect to LAN **32** and LAN **10**, a plurality of documents or resource objects may  
7 be stored within storage device **20** and controlled by mainframe computer **18**, as resource  
8 manager or library service for the resource objects thus stored. Of course, those skilled in the  
9 art will appreciate that mainframe computer **18** may be located a great geographic distance  
10 from LAN **10** and similarly, LAN **10** may be located a substantial distance from LAN **32**. For  
11 example, LAN **32** may be located in California while LAN **10** may be located within North  
12 Carolina and mainframe computer **18** may be located in New York.

13  
14 Software program code which employs the present invention is typically stored in the  
15 memory of a storage device **14** of a stand alone workstation or LAN server from which a  
16 developer may access the code for distribution purposes, the software program code may be  
17 embodied on any of a variety of known media for use with a data processing system such as a  
18 diskette or CD-ROM or may be distributed to users from a memory of one computer system  
19 over a network of some type to other computer systems for use by users of such other systems.  
20 Such techniques and methods for embodying software code on media and/or distributing  
21 software code are well-known and will not be further discussed herein.

22  
23 As will be appreciated upon reference to the foregoing, it is often desirable for a user  
24 to link an application program on the mainframe **18** to the internet **32** and/or World Wide Web  
25 (Web), where the application program was not originally designed for Web or internet based  
26 transactions. For such Web-enablement, the user needs to identify call interfaces of the  
27 application program which conventionally is a manual process. If this application program has  
28 a database dependency, then it may also be desirable for this user to identify call interfaces or  
29 computer program code which may be re-implemented as a database stored procedure call. A

1 preferred embodiment of the present invention assists a user in performing such an  
2 identification of computer program code which is a candidate for a particular purpose, such as  
3 re-implementation. Alternative embodiments identify candidates for re-implementation such as  
4 Web enablement or a stored procedure.  
5

6 Referring now to **Figure 2** illustrating a block diagram of a preferred embodiment of  
7 the present invention, the preferred embodiment comprises a database **215** (Application  
8 Knowledge Base) for storing static information **220**, dynamic information **225**, and  
9 relationships and dependencies **250** corresponding to computer code in application inventory  
10 **205**. The database in the preferred embodiment is an IBM® DB2® UDB (Universal  
11 Database); however, other relational or non-relational (object, network, hierarchal, flat, . . . )  
12 databases may be used in addition to either persistent or in-memory databases without  
13 departing from the spirit and scope of the present invention. (IBM® and DB2® are registered  
14 trademarks of International Business Machines Corporation in the United States, other  
15 countries, or both.) The static information **220** concerning the computer program code **205** is  
16 determined by a scanner, parser, or other software analysis tool **210**. An execution monitor or  
17 scanner **230** collects dynamic information **225** concerning the computer program code **205**  
18 during an execution **235** of the computer program code. The static information **220** and the  
19 dynamic information **225** are analyzed by an relationship analyzer **245** to discover and/or  
20 develop relationships and dependencies **250** responsive to the static information **220** and  
21 dynamic information **225**. A query **255** of the Application Knowledge Base **215** may then  
22 produce a set of potential candidates **260** of computer program code meeting a constraint of the  
23 query. The query facility of the preferred embodiment is IBM net.Data, a dynamic HTML  
24 (HyperText Markup Language) page generator; however, other query facilities may be used  
25 such as a Java servlet or a Java data bean. (Java and all Java-based trademarks are trademarks  
26 of Sun Microsystems, Inc. in the United States, other countries, or both.)  
27

28 **Figure 3** illustrates a preferred embodiment **300** of the static scanner, parser, or other  
29 analysis tool **210** which collects the static information **220** concerning the computer program

code **205**. The static scanner **300** may comprise multiple static analysis tools. In the preferred embodiment, the static scanner **300** comprises a control/environment parsing tool **305**, a transaction parsing tool **310**, a program source parsing tool **315**, a database parsing tool **320**, and an executable/ load module parsing tool **325**. The preferred control/environment parsing tool **305** is a JCL (Job Control Language) parsing tool which analyzes JCL **330** to provide program execution, dataset creation and use, and database use **335**. An IBM Customer Information Control System (CICS®) parsing tool and an IBM Information Management System (IMS) parsing tool are preferred transaction parsing tools **310** providing program execution, dataset and database use **340** from analyzing transaction definitions and file definitions **345**. (CICS® is a registered trademark of International Business Machines Corporation in the United States, other countries, or both.) Application program source **350** may be analyzed by the program source parsing tool **315** to provide data record definitions, data store I/O, call relationships, and data element use **355**. A database parsing tool for IMS DBD (Data Base Definition) and IMS PSB (Program Specification Block) and a database parsing tool to analyze DB2 table information are preferred for the database parsing tool **320** which analyzes database definitions **360** to provide linkages **365** between the application program source and databases such as IMS, PSB, and DB2 among others. The executable/load module parsing tool **325** analyzes load modules **370** to provide module, CSECT (control section), translator, and translation date **375**.

**Figure 4** illustrates a preferred embodiment **400** of the execution monitor or dynamic scanner **230** which collects dynamic information **225** concerning the computer program code **205** during an execution **235** of the computer program code. The dynamic scanner **400** may also comprise multiple dynamic analysis tools, comprising in the preferred embodiment a coverage and performance measurement tool **410** and an inter-dependency tool **420**. The coverage and performance measurement tool **410** analyzes the execution **430** of an application program run unit and compilation unit to determine the coverage for single or multiple test cases **440** and to measure performance **450**. A preferred inter-dependency tool **420** is a CICS Inter-dependencies Utility which dynamically captures resource usage, transaction flow **460**

1 from a CICS region **470**. The execution monitor in the preferred embodiment is an IBM  
2 CICS Affinities Utility and the IBM CICS Interdependency Utility; however, those skilled in  
3 the art will recognize other execution monitors may be used without departing from the spirit  
4 and scope of the invention.

5  
6 **Figure 5** illustrates a preferred embodiment **500** of the relationship analyzer **245** which  
7 discovers and/or develops relationships and dependencies **250** responsive to the static  
8 information **220** and dynamic information **225**. In the preferred embodiment, the relationship  
9 analyzer **500** provides automatic, generalized, inter-relationship analysis by performing  
10 IMS DL/I (Information Management System Data Language 1) call parameter analysis **510**,  
11 call hierarchy expansion **520**, load module to source relationship mapping **530**, CICS  
12 pseudo-conversation flow analysis **540**, missing and duplicate component analysis **550**, and un-  
13 referenced component analysis **560**.

14  
15 Referring now to **Figure 6**, the flowchart illustrates the operations preferred in carrying  
16 out the preferred embodiment of the present invention. In the flowcharts, the graphical  
17 conventions of a diamond for a test or decision and a rectangle for a process or function are  
18 used. These conventions are well understood by those skilled in the art, and the flowcharts are  
19 sufficient to enable one of ordinary skill to write code in any suitable computer programming  
20 language. The process **600** of identifying computer program code which is a candidate for a  
21 particular purpose, such as re-implementation for Web enablement or stored procedures begins  
22 at process block **605**. Process block **610** then uses static scanner **210** to scan or parse the  
23 application inventory **205** retaining static information **220** concerning the artifacts of the  
24 application program. Thereafter, process block **615** stores the static information **220** in the  
25 Application Knowledge Base **215**. Process block **620** augments the Application Knowledge  
26 Base **215** with dynamic information **225** obtained by an execution scanner or monitor **230**  
27 which collects the dynamic information **225** during the execution of an application program  
28 executable **235** produced by an assembler, compiler, or interpreter **240** from the computer  
29 program source code. After collection of the static information **220** and dynamic information

1     **225**, process block **625** uses relationship analyzer **245** to discover and develop relationships  
2     and dependencies **250** within the Application Knowledge Base **215**, and then stores these  
3     relationships and dependencies **250** in the Application Knowledge Base **215**. Thereafter,  
4     process block **630** may query **255** the application knowledge base **215** to produce a set of  
5     potential candidates **260** of computer program code meeting a constraint of the query.  
6

7             For example, to determine which applications may benefit by being re-implemented as  
8     a database stored procedure, the query may search for routines containing SQL (Structured  
9     Query Language) statements that are the targets of a number of CALLs where the number is  
10    above a specified threshold. Thus the present invention provides a capability of identifying  
11    more likely candidates as opposed to the conventional manual process of looking for routines  
12    that contain a large amount of SQL. To determine which application programs are candidates  
13    for a Web interface under CICS, the query may search for CICS transactions that contain a  
14    COMMAREA (communication area) linkage and that are void of BMS (Basic Mapping  
15    Support) MAP send and/or receive statements (i.e., that lack screen I/O (input output)).  
16

17            After the results of the query are returned, decision block **635** determines if sufficient  
18    candidates are identified and returned. If a sufficient number of candidates are not discovered,  
19    then process block **645** allows relaxing the query constraints, after which control returns to  
20    process block **630** to query the application knowledge base **215** with the relaxed constraints.  
21

22            Returning now to decision block **635**, if sufficient candidates are identified, the process  
23    may end at process block **640**.  
24

25            Using the foregoing specification, the invention may be implemented using standard  
26    programming and/or engineering techniques using computer programming software, firmware,  
27    hardware or any combination or sub-combination thereof. Any such resulting program(s),  
28    having computer readable program code means, may be embodied within one or more  
29    computer usable media such as fixed (hard) drives, disk, diskettes, optical disks, magnetic tape,



1 semiconductor memories such as Read-Only Memory (ROM), Programmable Read-Only  
2 Memory (PROM), etc., or any memory or transmitting device, thereby making a computer  
3 program product, i.e., an article of manufacture, according to the invention. The article of  
4 manufacture containing the computer programming code may be made and/or used by  
5 executing the code directly or indirectly from one medium, by copying the code from one  
6 medium to another medium, or by transmitting the code over a network. An apparatus for  
7 making, using, or selling the invention may be one or more processing systems including, but  
8 not limited to, central processing unit (CPU), memory, storage devices, communication links,  
9 communication devices, servers, input/output (I/O) devices, or any sub-components or  
10 individual parts of one or more processing systems, including software, firmware, hardware or  
11 any combination or sub-combination thereof, which embody the invention as set forth in the  
12 claims.

13  
14 User input may be received from the keyboard, mouse, pen, voice, touch screen, or any  
15 other means by which a human can input data to a computer, including through other programs  
16 such as application programs.

17  
18 One skilled in the art of computer science will easily be able to combine the software  
19 created as described with appropriate general purpose or special purpose computer hardware to  
20 create a computer system and/or computer sub-components embodying the invention and to  
21 create a computer system and/or computer sub-components for carrying out the method of the  
22 invention. Although the present invention has been particularly shown and described with  
23 reference to a preferred embodiment, it should be apparent that modifications and adaptations  
24 to that embodiment may occur to one skilled in the art without departing from the spirit or  
25 scope of the present invention as set forth in the following claims.

## CLAIMS

We claim:

1. A method of identifying a portion of computer program code as a candidate for a modification, said method comprising the steps of :
  - scanning and parsing source code corresponding to the computer program code to determine static information concerning the computer program code;
  - storing the static information in a database;
  - collecting dynamic information concerning the computer program code during an execution of the computer program code;
  - storing the dynamic information in the database;
  - developing relationships and dependencies responsive to the static information and dynamic information stored in database;
  - storing the relationships and dependencies in the database; and
  - providing for a query of the database to produce a set of potential candidates of computer program code meeting a constraint of the query.
2. The method of claim 1 wherein if insufficient candidates are returned by the query, then providing for a relaxing of the constraint, and a repetition of the query.
3. The method of claim 2 further comprising the step of identifying a portion of the computer program code as a candidate for re-implementation as a database stored procedure call.

1 4. The method of claim 3 wherein the query searches the database for a portion of the  
2 computer program code having static information indicating that the portion of the computer  
3 program code contains a call to a database management system, and having dynamic  
4 information indicating that the portion of the computer program code is subject to a call by  
5 another portion of computer program code.

1 5. The method of claim 4 wherein the query searches the database for a portion of the  
2 computer program code having static information indicating that the portion of the computer  
3 program code contains a number, above a specified first threshold, of calls to a database  
4 management system, and having dynamic information indicating that the portion of the  
5 computer program code is subject to a number of calls, above a specified second threshold, by  
another portion of computer program code.

6. The method of claim 2 further comprising the step of identifying a portion of the  
computer program code as a candidate for re-implementation as a Web-enabling interface call.

7. The method of claim 6 wherein the query searches the database for a portion of the  
computer program code having static information indicating that the portion of the computer  
program code contains a transaction call.

1 8. The method of claim 7 wherein the query also searches the database for a portion of the  
2 computer program code having static information indicating that the portion of the computer  
3 program does not contain screen output related program code.

1 9. An article of manufacture for use in a computer system for identifying a portion of  
2 computer program code as a candidate for a modification, said article of manufacture  
3 comprising a computer-readable storage medium having a computer program embodied in said  
4 medium which causes the computer system to execute a method comprising the steps of:

5 scanning and parsing source code corresponding to the computer program code  
6 to determine static information concerning the computer program code;

7 storing the static information in a database;

8 collecting dynamic information concerning the computer program code during  
9 an execution of the computer program code;

10 storing the dynamic information in the database;

11 developing relationships and dependencies responsive to the static information  
12 and dynamic information stored in database;

13 storing the relationships and dependencies in the database; and

14 providing for a query of the database to produce a set of potential candidates of  
15 computer program code meeting a constraint of the query.

16 10. The article of manufacture of claim 9 wherein if insufficient candidates are returned by  
17 the query, then providing for a relaxing of the constraint, and a repetition of the query.

1 11. The article of manufacture of claim 10 wherein the embodied computer program  
2 causes the computer system to execute the method step of identifying a portion of the  
3 computer program code as a candidate for re-implementation as a database stored procedure  
4 call.

12. The article of manufacture of claim 11 wherein the query searches the database for a portion of the computer program code having static information indicating that the portion of the computer program code contains a call to a database management system, and having dynamic information indicating that the portion of the computer program code is subject to a call by another portion of computer program code.

13. The article of manufacture of claim 12 wherein the query searches the database for a portion of the computer program code having static information indicating that the portion of the computer program code contains a number, above a specified first threshold, of calls to a database management system, and having dynamic information indicating that the portion of the computer program code is subject to a number of calls, above a specified second threshold, by another portion of computer program code.

14. The article of manufacture of claim 10 wherein the embodied computer program causes the computer system to execute the method step of identifying a portion of the computer program code as a candidate for re-implementation as a Web-enabling interface call.

15. The article of manufacture of claim 14 wherein the query searches the database for a portion of the computer program code having static information indicating that the portion of the computer program code contains a transaction call.

16. The article of manufacture of claim 15 wherein the query also searches the database for a portion of the computer program code having static information indicating that the portion of the computer program does not contain screen output related program code.

1 17. A computer system for identifying a portion of computer program code as a candidate  
2 for a modification, said computer system comprising :

3 a parser for scanning and parsing source code corresponding to the computer  
4 program code to determine static information concerning the computer program code;

5 an execution monitor for collecting dynamic information concerning the  
6 computer program code during an execution of the computer program code;

7 relationships and dependencies developed responsive to the static information  
8 and dynamic information;

9 a database for storing the static information, the dynamic information, and the  
10 relationships and dependencies; and

11 a query of the database to produce a set of potential candidates of computer  
12 program code meeting a constraint of the query.

1 18. The computer system of claim 17 wherein if insufficient candidates are returned by the  
2 query, then providing for a relaxing of the constraint, and a repetition of the query.

1 19. The computer system of claim 18 further comprising an identification of a portion of  
2 the computer program code as a candidate for re-implementation as a database stored procedure  
3 call.

1 20. The computer system of claim 19 wherein the query searches the database for a portion  
2 of the computer program code having static information indicating that the portion of the  
3 computer program code contains a call to a database management system, and having dynamic  
4 information indicating that the portion of the computer program code is subject to a call by  
5 another portion of computer program code.

1 21. The computer system of claim 20 wherein the query searches the database for a portion  
2 of the computer program code having static information indicating that the portion of the  
3 computer program code contains a number, above a specified first threshold, of calls to a  
4 database management system, and having dynamic information indicating that the portion of  
5 the computer program code is subject to a number of calls, above a specified second threshold,  
6 by another portion of computer program code.

1 22. The computer system of claim 18 further comprising an identification of a portion of  
2 the computer program code as a candidate for re-implementation as a Web-enabling interface  
3 call.

1 23. The computer system of claim 22 wherein the query searches the database for a portion  
2 of the computer program code having static information indicating that the portion of the  
3 computer program code contains a transaction call.

1 24. The computer system of claim 23 wherein the query also searches the database for a  
2 portion of the computer program code having static information indicating that the portion of  
3 the computer program does not contain screen output related program code.

## ABSTRACT

### **METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR PROVIDING AUTOMATIC IDENTIFICATION OF A COMPUTER PROGRAM CODE CANDIDATE FOR WEB DEPLOYMENT OR A STORED PROCEDURE**

Computer program code which is a candidate for Web enablement or stored procedures is identified. Source code corresponding to computer program code is scanned and parsed to determine static information concerning the computer program code. The static information is stored in a database. Dynamic information concerning the computer program code during an execution of the computer program code is also collected and stored in the database. Responsive to the static information and dynamic information stored in the database, relationships and dependencies are then developed and stored in the database. The database may then be queried to produce a set of potential candidates of computer program code meeting a constraint of the query. If insufficient candidates are returned by the query, then the query constraint may be relaxed, and the query repeated. A candidate for re-implementation as a database stored procedure call may be identified by a query searching the database for a portion of the computer program code having static information indicating that the portion of the computer program code contains a number, above a specified first threshold, of calls to a database management system, and having dynamic information indicating that the portion of the computer program code is subject to a number of calls, above a specified second threshold, by another portion of computer program code. A candidate for re-implementation as a Web-enabling interface call may be identified by a query searching the database for a portion of the computer program code having static information indicating that the portion of the computer program contains a transaction and does not contain screen output related program code.





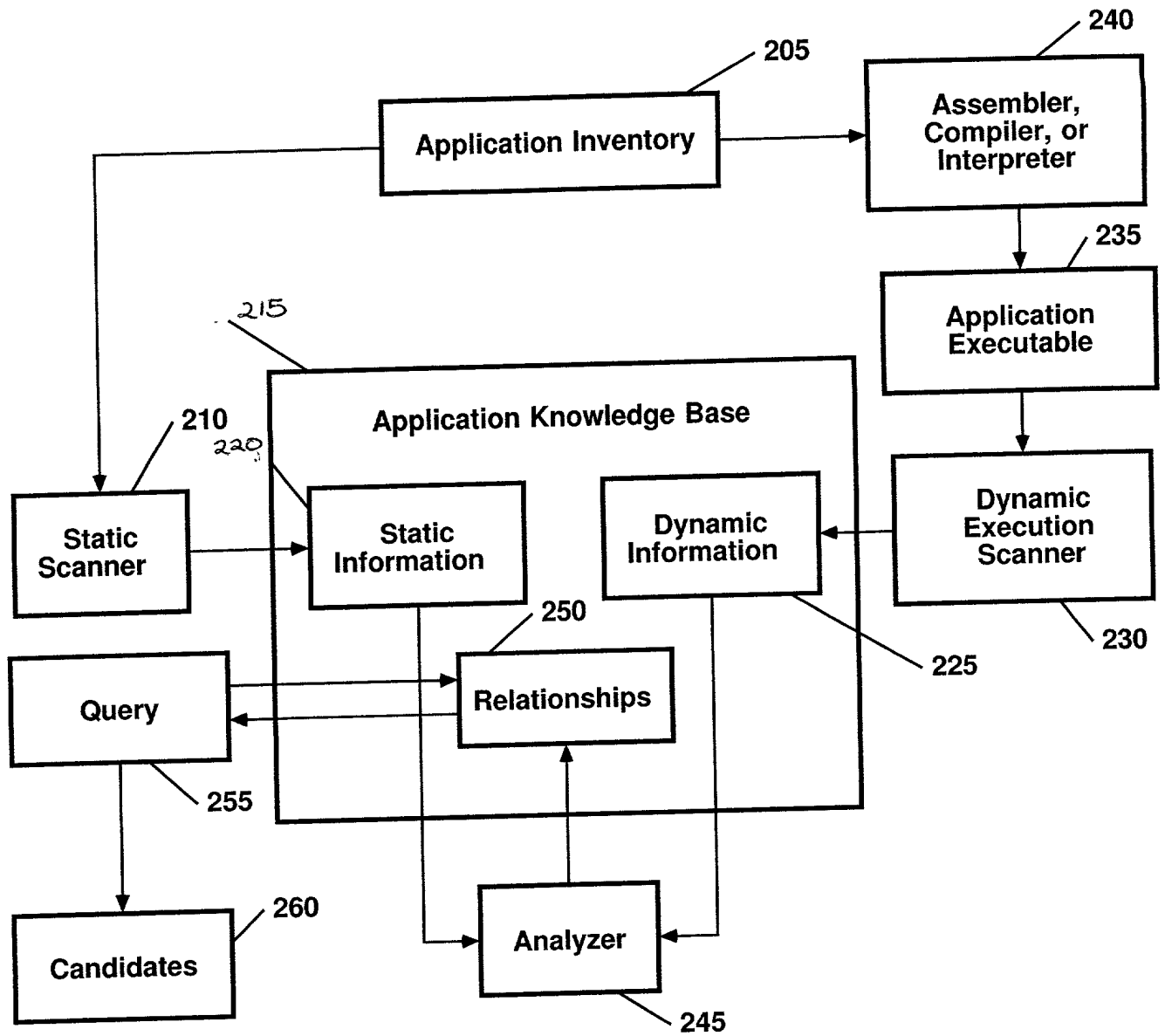
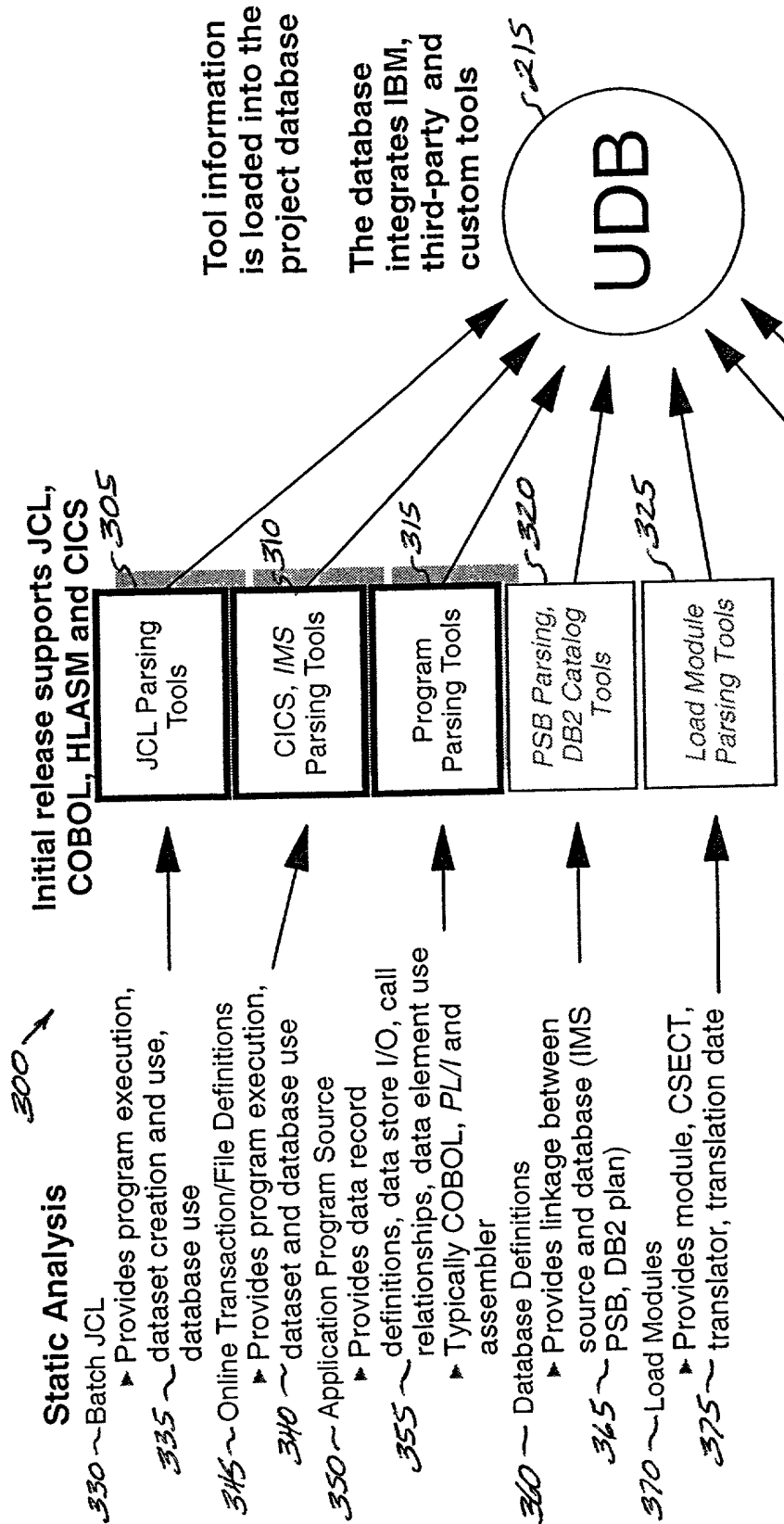


Fig. 2



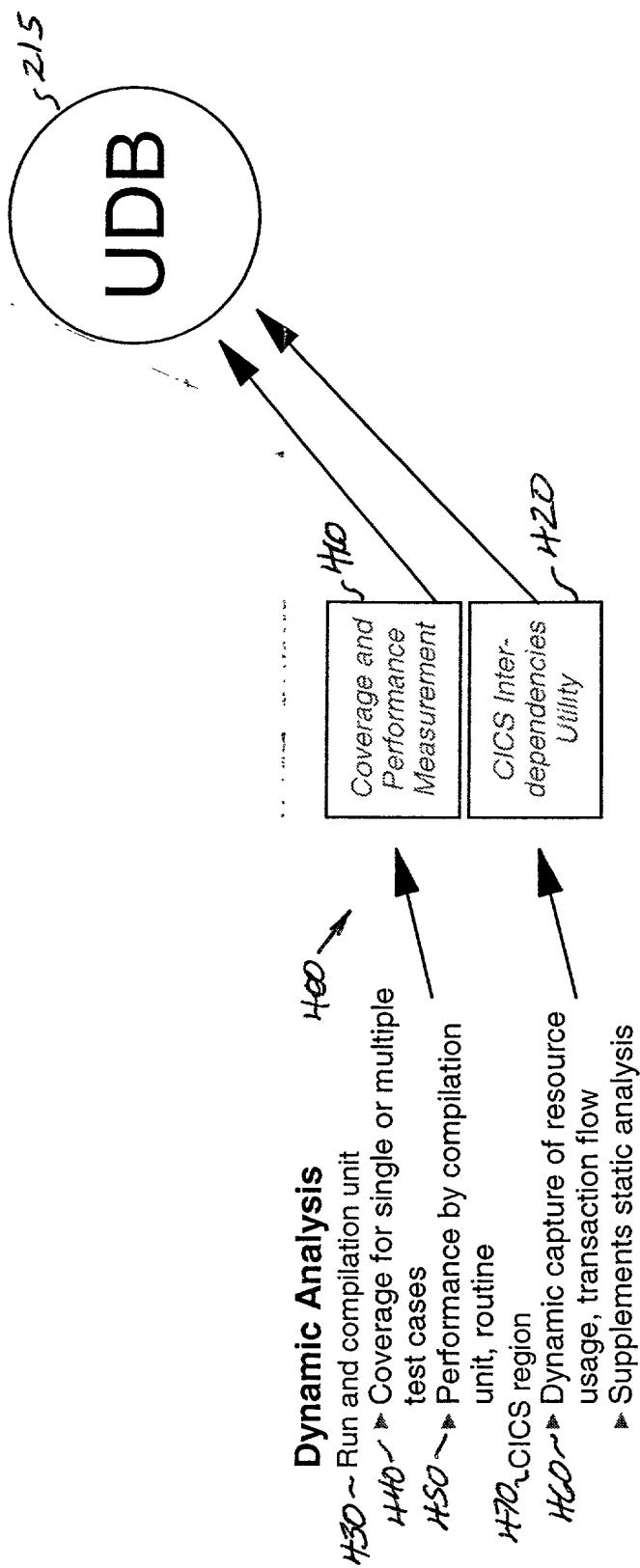
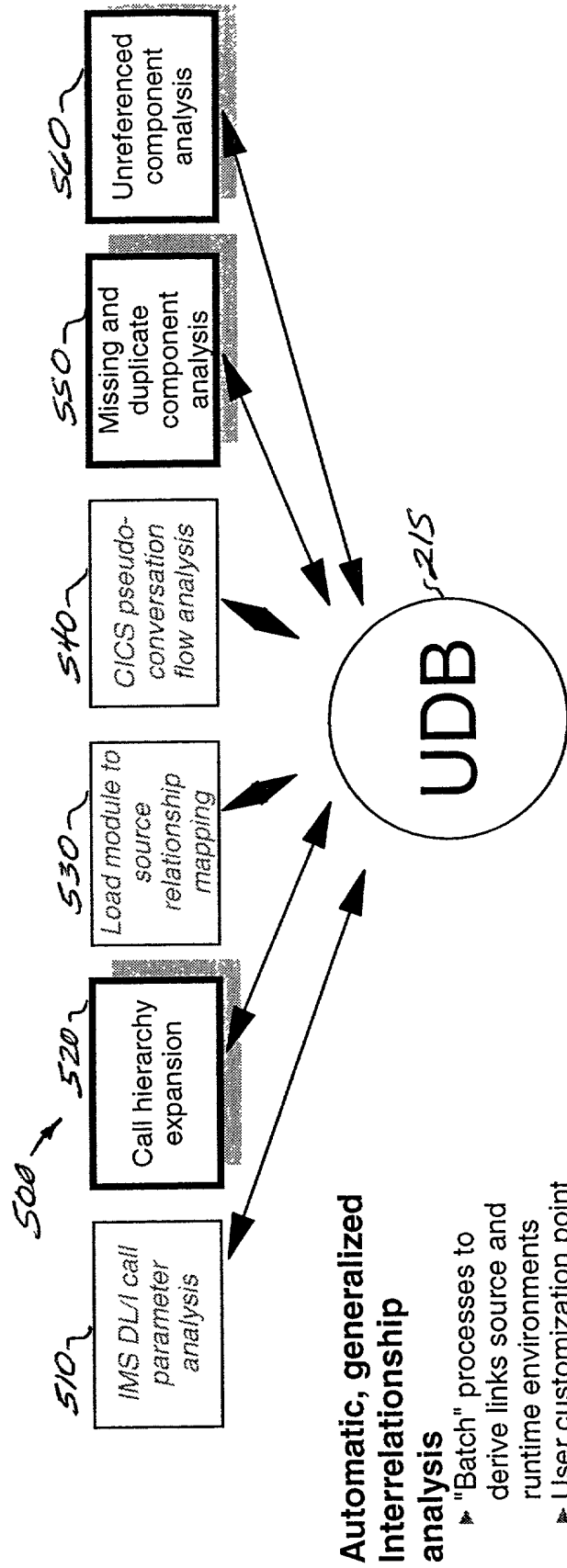


Fig. 4



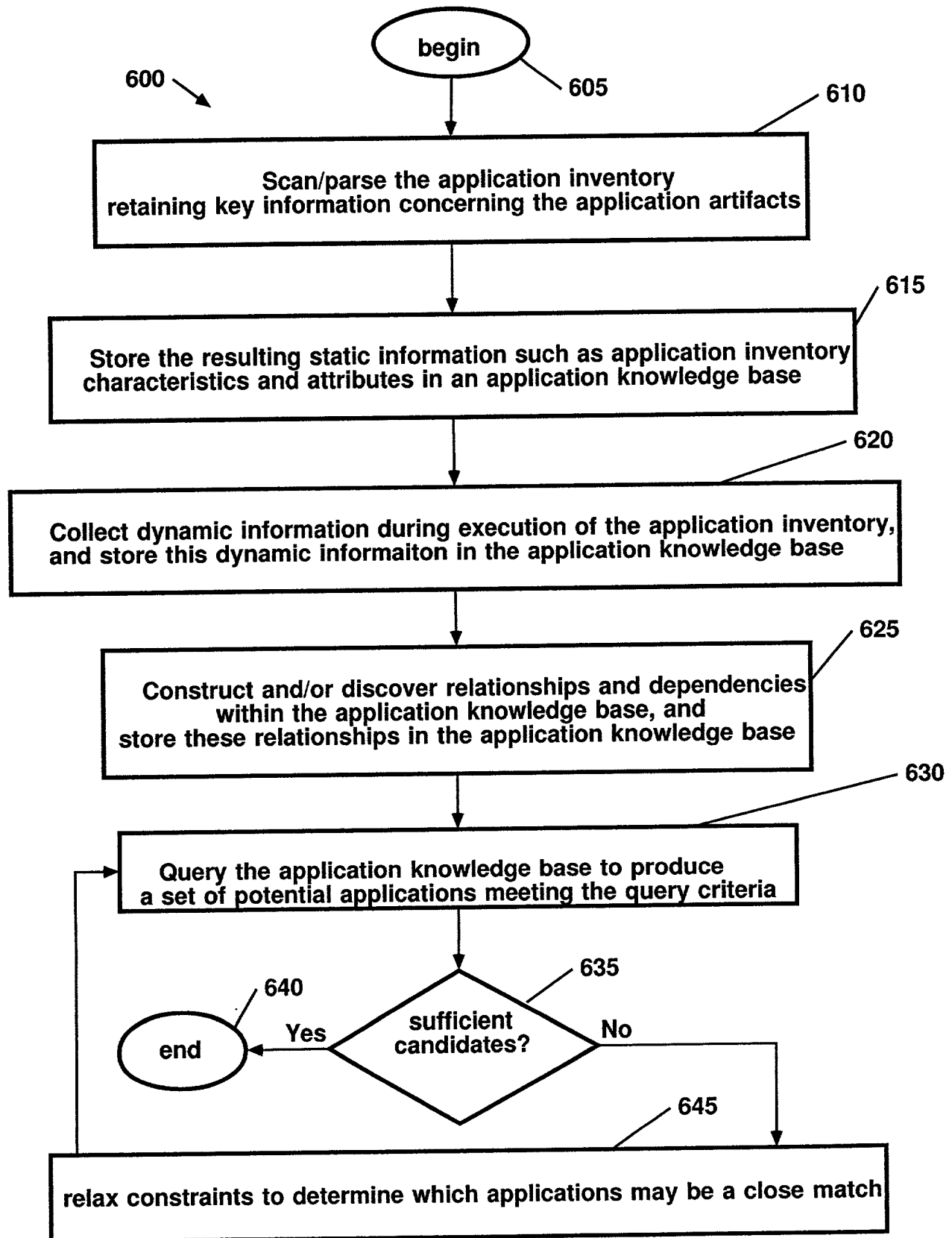


Fig. 6

## DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

DOCKET: STL9-2000-0080US1

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

Method of, System for, and Computer Program Product for Providing Automatic Identification of a Computer Program Code Candidate for Web Deployment or a Stored Procedure

the specification of which (check one)

☒ is attached hereto.  
☐ was filed on \_\_\_\_\_  
as Application Serial No. \_\_\_\_\_  
and was amended on \_\_\_\_\_ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)	Priority Claimed
<input type="checkbox"/> None (Number)	<input type="checkbox"/> Yes <input type="checkbox"/> No
(Country)	
(Day/Month/Year Filed)	

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56, which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

<input type="checkbox"/> None (Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)
-----------------------------------------------------------	---------------	-----------------------------------------

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

DOCKET: STL9-2000-0080US1

**POWER OF ATTORNEY:** As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (list name and registration number)

Romualdas Strimaitis (#35,697)	Christopher A. Hughes (#26,914)
Timothy M. Farrell (#37,321)	Edward A. Pennington (#32,588)
Ingrid M. Foerster (#36,511)	John E. Hoel (#26,279)
Prentiss W. Johnson (# 33,123)	Joseph C. Redmond, Jr. (#18,753)

**Send correspondence to:**

Prentiss Wayne Johnson  
Attorney  
IBM Corporation  
Intellectual Property Law  
555 Bailey Avenue (J46/G467)  
San Jose CA 95141-1003

**Direct Telephone Calls to:** (name and telephone number) Prentiss W. Johnson, 408-463-5673

**Full name of sole or first joint-inventor:** Curt L. Cotner

Inventor's signature:

Date:

Residence: 1975 Day Road, Gilroy, California 95020

Citizenship: United States of America

Post Office Address: Same

**Full name of second joint-inventor:** Laurence E. England

Inventor's signature:

Date:

Residence: 520 La Canada Court, Morgan Hill, California 95037

Citizenship: United States of America

Post Office Address: Same

**Full name of third joint-inventor:** Howard J. Glaser

Inventor's signature:

Date:

Residence: 5808 Vargas Court, San Jose, California 95120

Citizenship: United States

Post Office Address: Same



DOCKET: STL9-2000-0080US1

Inventor's signature:

Date:

Residence: 3033 Thayer Street, Evanston, IL 60201-1146

Citizenship: **United States**

Post Office Address: Same

